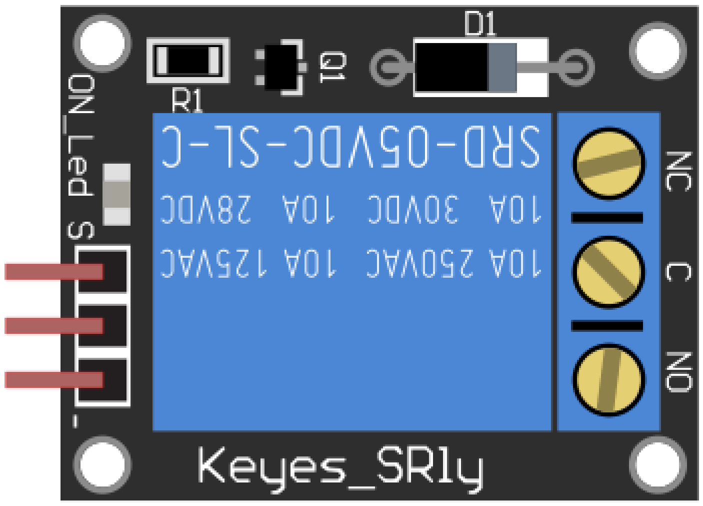
Relay Module

A relay is basically a switch which is operated by an electromagnet. The electromagnet requires a small voltage to get activated which we will give from the Arduino and once it is activated, it will pull the contact to make the high voltage circuit.

The relay module we are going to use is the SRD-05VDC-SL-C. It runs on 5V and we can control it with any micro-controller but we are going to use Arduino.

Pin out of 5V relay module



Relay Module Pin out

The Arduino relay module has total of six pins: three on one side and three on other side.On the bottom side, there are three pins which are signal, 5V and ground. We will connect these pins with the Arduino. While on the other side, there are NC (Normally close), C (Common) and the NO (normally open) which are the output pins of the 5V relay. There, we will connect the output device.

Normally open state (NO) VS Normally closed state (NC)

The Arduino relay module can be used in two states which are

1. Normally open state (NO)

2. Normally closed state (NC)

Normally open (NO)

In the normally open state, the initial output of the relay will be low when it will be powered. In this state, the common and the normally open pins are used.

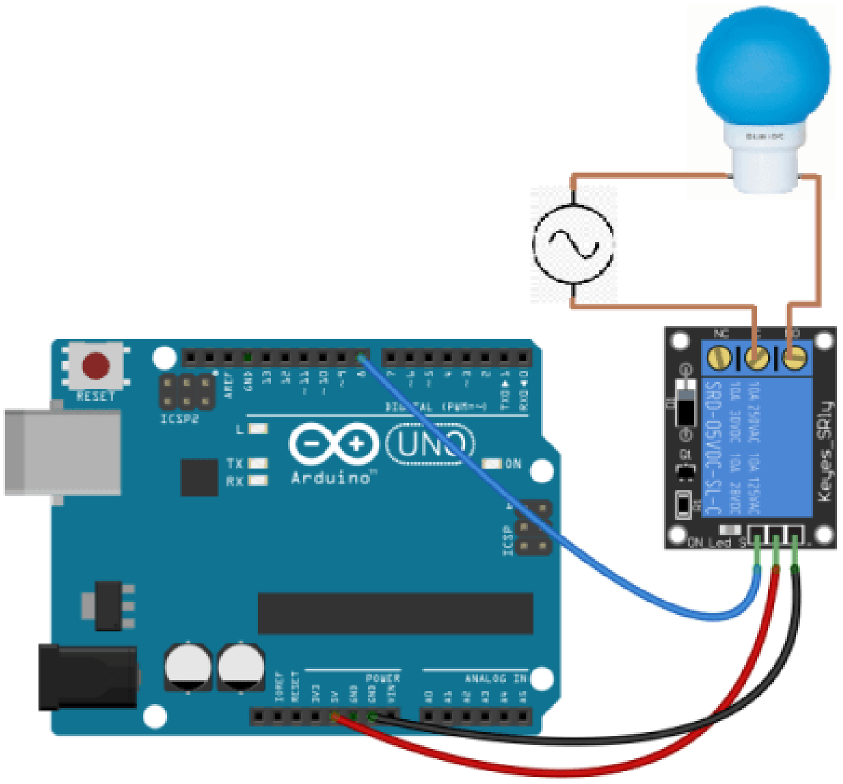
Normally closed state (NC)

In the normally closed state, the initial output of the relay will be high when it will be powered. In this state, the common and the normally close pins are used.

Controlling AC Device using Arduino Relay Module

For the control of AC device, you need to take the necessary precautions because the AC is dangerous and it can cause damage to you. So, to avoid any danger, follow the below tutorial correctly.

Circuit Diagram and Explanation



For the control of AC device, we will require an external source which will power the AC source. So, connect the VCC, ground and signal to the 5V, ground and pin 8 of Arduino respectively. On the other end, connect one wire of the AC source to the one end of the bulb and the other wire to the common (C) of the relay. Then connect the normally open (NO) to the other end of the bulb.

NodeMCU

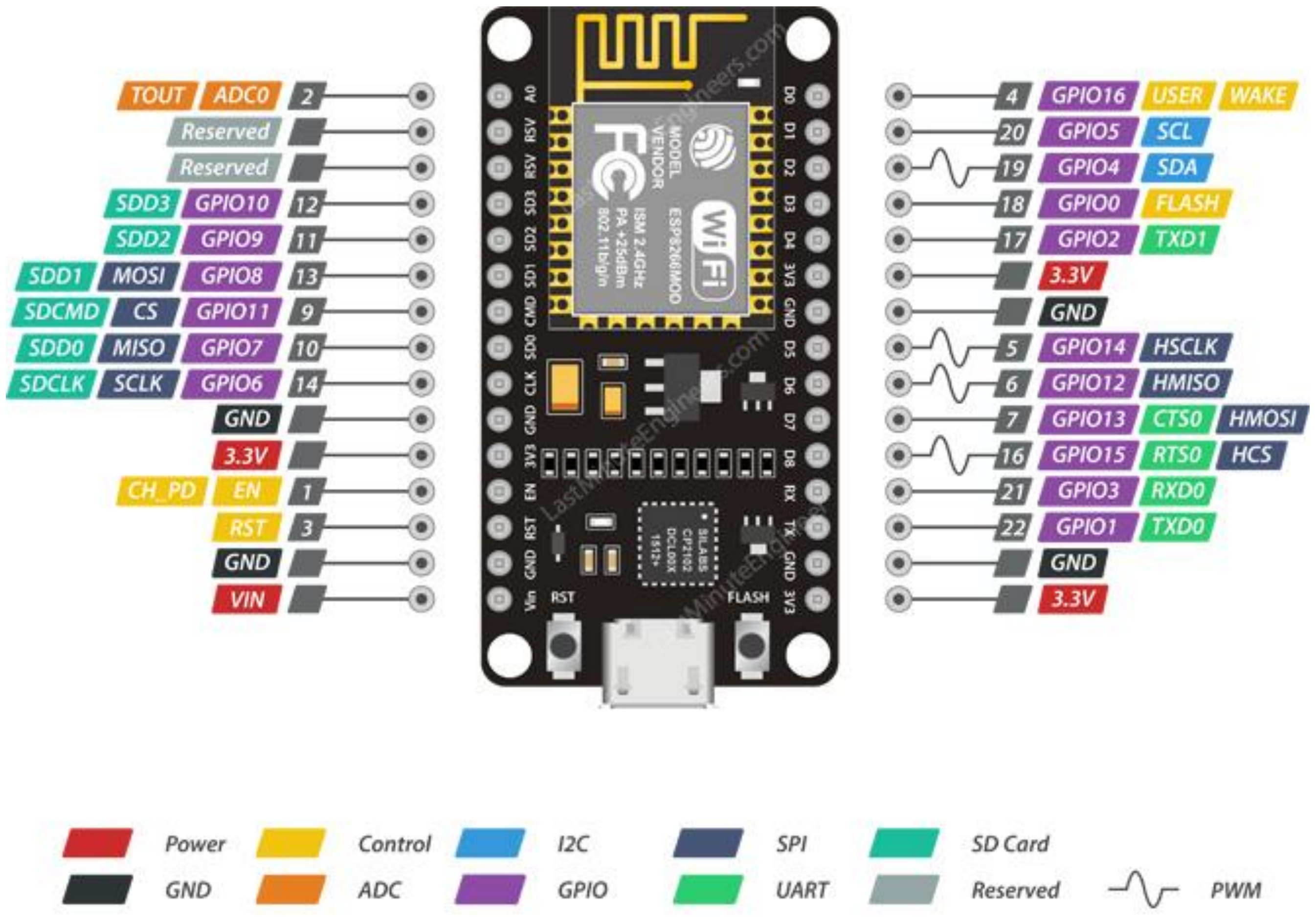
NodeMCU is an open source IoT platform. It includes firmware which runs on the ESP8266 Wi-Fi SoC from Espressif Systems, and hardware which is based on the ESP-12 module. The term "NodeMCU" by default refers to the firmware rather than the development kits. The firmware uses the Lua scripting language. It is based on the eLua project, and built on the Espressif Non-OS SDK for ESP8266. It uses many open source projects, such as lua-cjson.

ESP8266 is a low-cost, wifi module chip that can be configured to connect to the internet for IOT and similar technology projects. Basically your normal electrical & mechanical equipment cannot connect to the internet their own.

- The device features 4MB of flash **memory**, 80MHz of system clock, around 50k of usable **RAM** and an on chip Wifi Transceiver.
- The **NodeMCU** uses the SPX3819 3.3v **voltage** regulator
- It has got Micro USB slot that can be directly connected to the computer or other USB host devices. ... It has got **CP2102** USB to serial converter.

ESP8266 NodeMCU Pinout

The ESP8266 NodeMCU has total 30 pins that interface it to the outside world. The connections are as follows:



ESP8266 is a system on a chip (SoC) design with components like the processor chip. The processor has around 16 GPIO lines, some of which are used internally to interface with other components of the SoC, like flash memory.

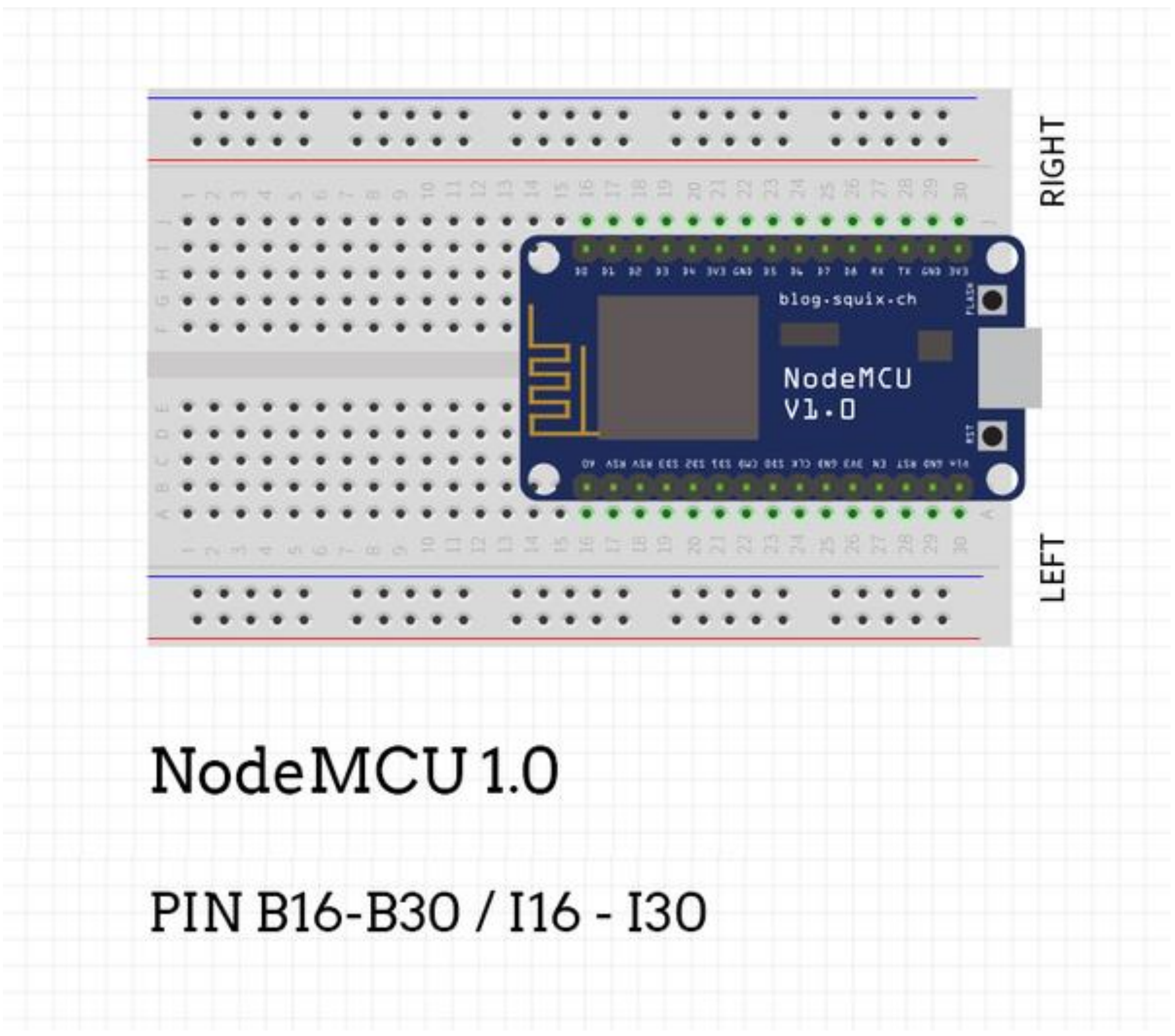
Since several lines are used internally within the ESP8266 SoC, we have about 11 GPIO pins remaining for GPIO purpose.

Now again 2 pins out of 11 are generally reserved for RX and TX in order to communicate with a host PC from which compiled object code is downloaded.

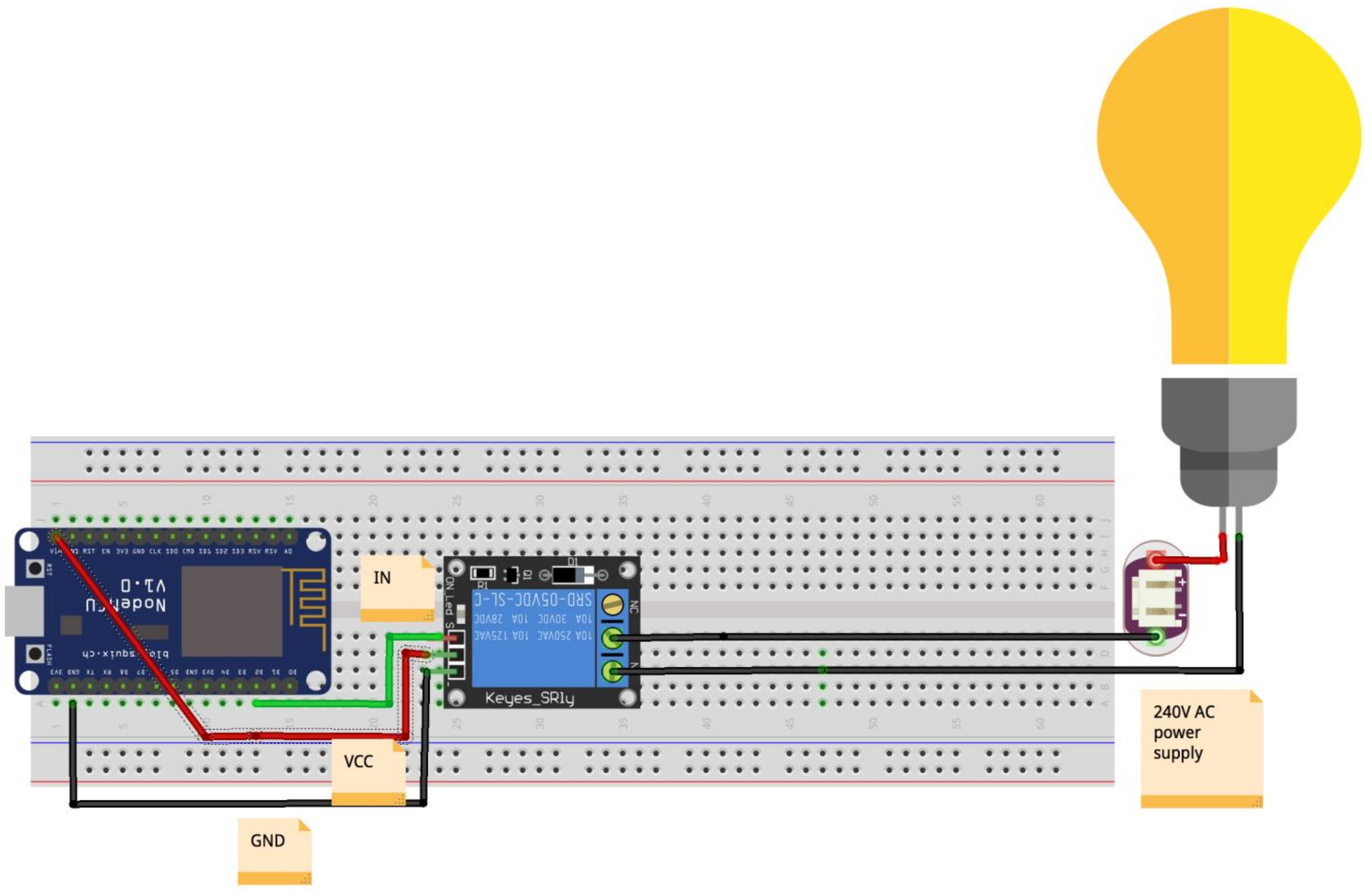
Hence finally, this leaves just 9 general purpose I/O pins i.e. D0 to D8.

As shown in above figure of NodeMCU Dev Kit. We can see RX, TX, SD2, SD3 pins are not mostly used as GPIOs since they are used for other internal process. But we can try with SD3 (D12) pin which mostly like to respond for GPIO/PWM/interrupt like functions.

Note that D0/GPIO16 pin can be only used as GPIO read/write, no special functions are supported on it.



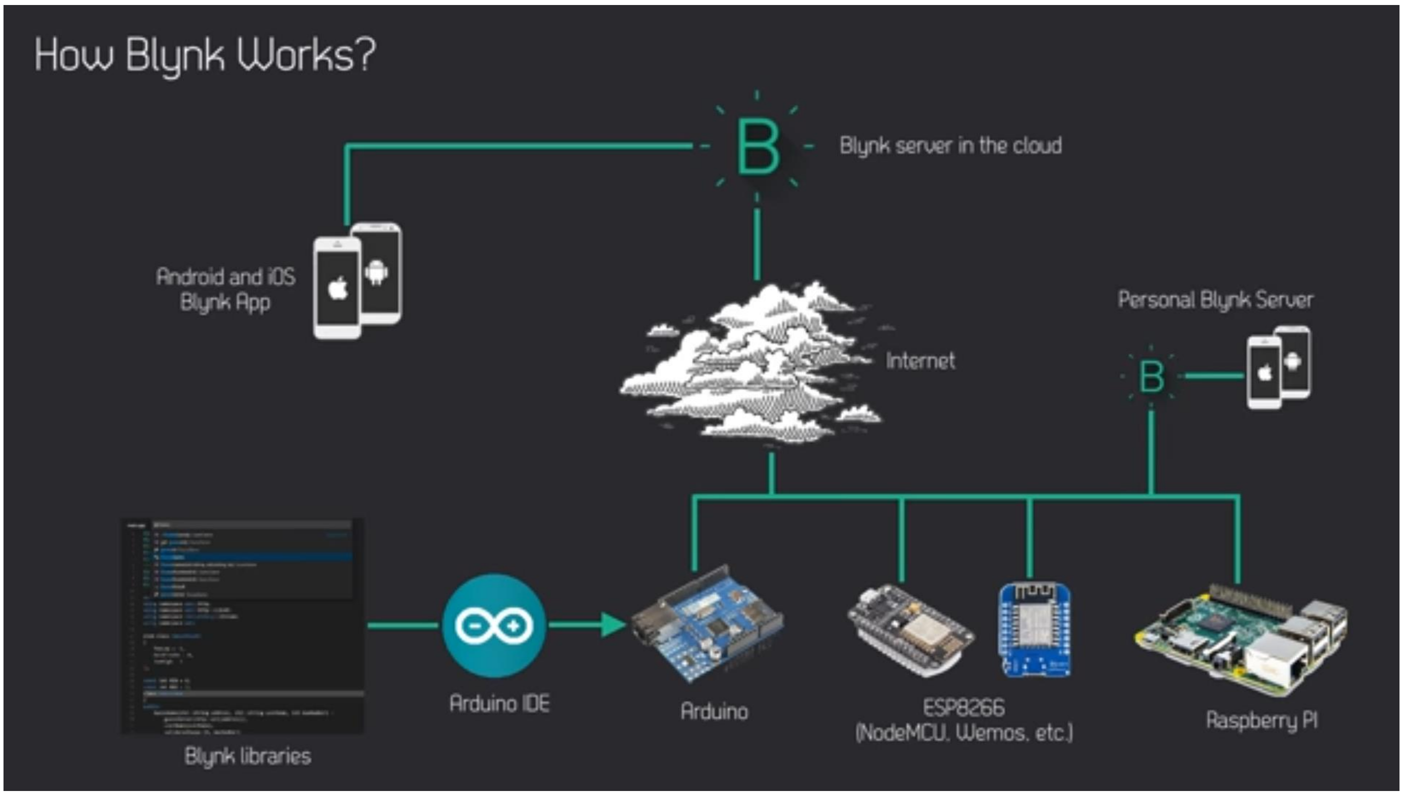
CIRCUIT DIAGRAM



<u>What is blynk and how does it work?</u>

Blynk is a Platform with IOS and Android apps to control Arduino, Raspberry Pi and the likes over the Internet. It's a digital dashboard where you can build a graphic interface for your project by simply dragging and dropping widgets.

How **Blynk Works**. **Blynk** was designed for the Internet of Things. It **can** control hardware remotely, it **can** display sensor data, it **can** store data, vizualize it and **do** many other cool things. ... **Blynk** Server - responsible for all the communications between the smartphone and hardware.



CODE

```
/*************************************************************
  Download latest Blynk library here:
    https://github.com/blynkkk/blynk-library/releases/latest

  Blynk is a platform with iOS and Android apps to control
  Arduino, Raspberry Pi and the likes over the Internet.
  You can easily build graphic interfaces for all your
  projects by simply dragging and dropping widgets.

    Downloads, docs, tutorials: http://www.blynk.cc
    Sketch generator:           http://examples.blynk.cc
    Blynk community:            http://community.blynk.cc
    Follow us:                  http://www.fb.com/blynkapp
                                http://twitter.com/blynk_app

  Blynk library is licensed under MIT license
  This example code is in public domain.

 *************************************************************
  This example runs directly on ESP8266 chip.

  Note: This requires ESP8266 support package:
    https://github.com/esp8266/Arduino

  Please be sure to select the right ESP8266 module
  in the Tools -> Board menu!

  Change WiFi ssid, pass, and Blynk auth token to run :)
  Feel free to apply it to any other example. It's simple!
 *************************************************************/

/* Comment this out to disable prints and save space */
#define BLYNK_PRINT Serial


#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>

// You should get Auth Token in the Blynk App.
// Go to the Project Settings (nut icon).
char auth[] = "NsMD7Q4MeUPAwVcuJ6JZdjWo5Z38s7GB";

// Your WiFi credentials.
// Set password to "" for open networks.
char ssid[] = "G";
char pass[] = "";

void setup()
{
  // Debug console
  Serial.begin(9600);

  Blynk.begin(auth, ssid, pass);
}

void loop()
{
  Blynk.run();
}
```

WHAT IS AUTH TOKEN IN CODE?

Every **Blynk** project is assigned a unique 32-byte string called the "**auth** token", which allows your hardware communicate with a specific project in the app. When you upload a new program to the **Blynk** Board, you need to program the **auth** token into it on top of any other **code** you may want to add.

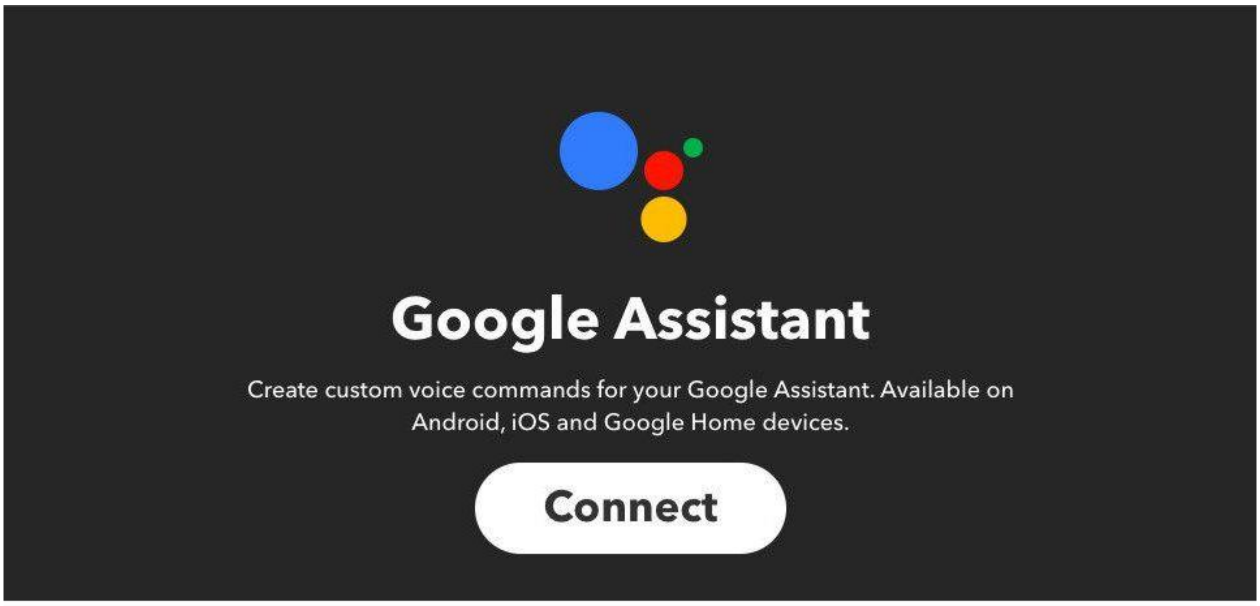
WHAT IS DONE BY blynk.begin() ?

Blynk.begin() function establishes network connection,sets up blynk connection and tries to connect to blynk server.

If This Then That, also known as **IFTTT** .It is a free web-based service to create chains of simple conditional statements, called *applets*.

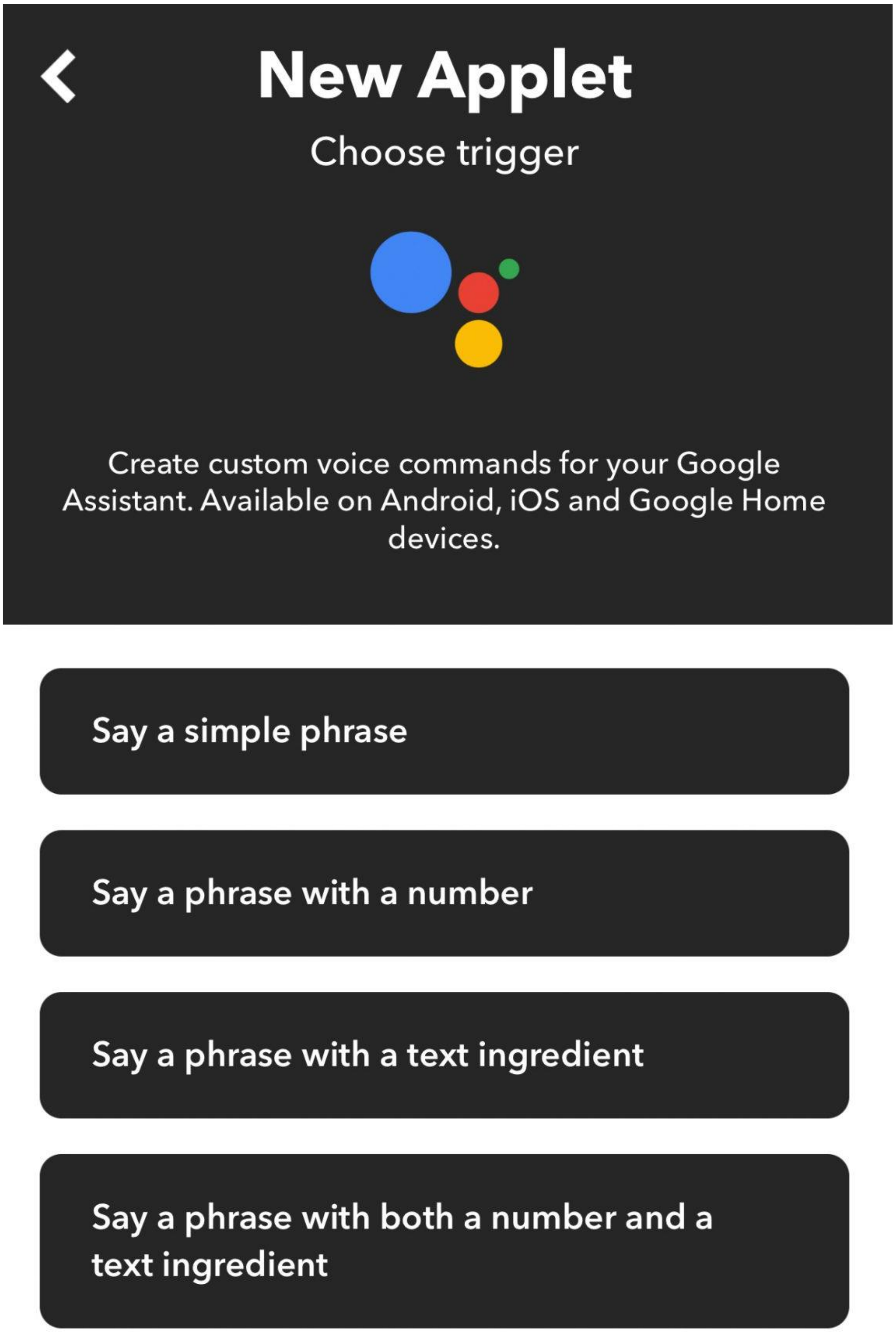
An applet is triggered by changes that occur within other web services such as Gmail, Facebook, Telegram, Instagram, or Pinterest.

CONNECT GOOGLE ASSISTANT TO IFTTT



Google Assistant

Create custom voice commands for your Google Assistant. Available on Android, iOS and Google Home devices.



New Applet

Choose trigger

Create custom voice commands for your Google Assistant. Available on Android, iOS and Google Home devices.

Say a simple phrase

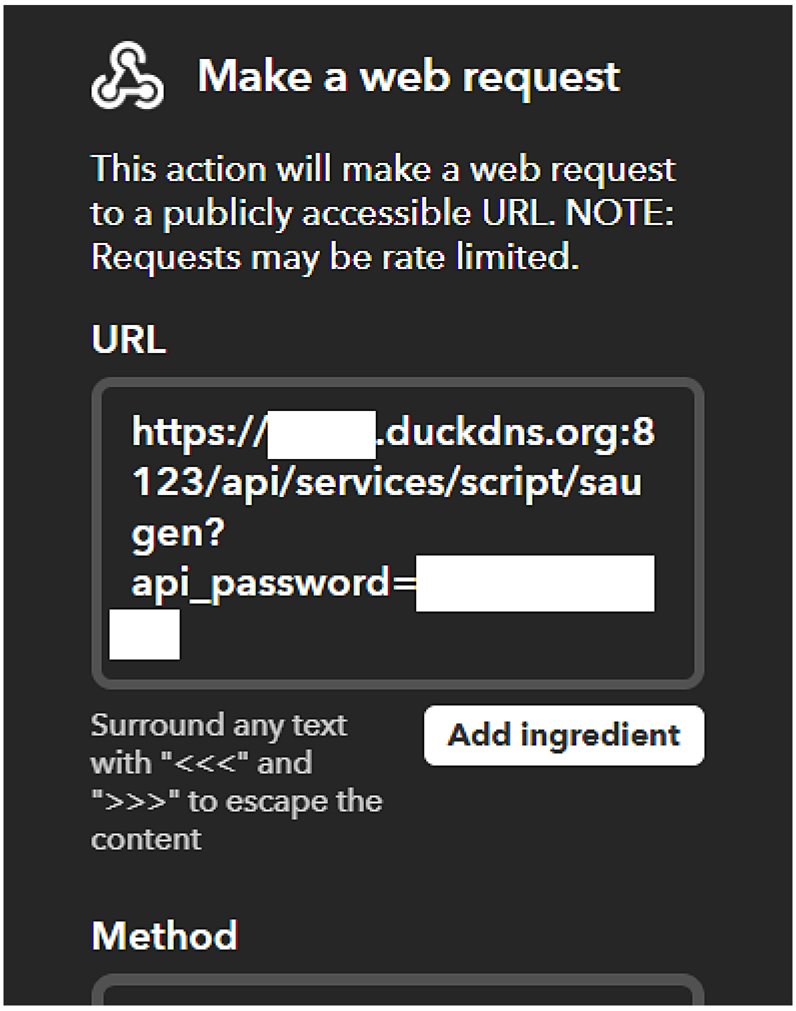
Say a phrase with a number

Say a phrase with a text ingredient

Say a phrase with both a number and a text ingredient

MAKE A WEB REQUEST WITH THE HELP OF WEBHOOKS

- A **webhook** (also called a web callback or HTTP push API) is a way for an app to provide other applications with real-time information. A **webhook** delivers data to other applications as it happens, meaning you get data immediately.



Make a web request

This action will make a web request to a publicly accessible URL. NOTE: Requests may be rate limited.

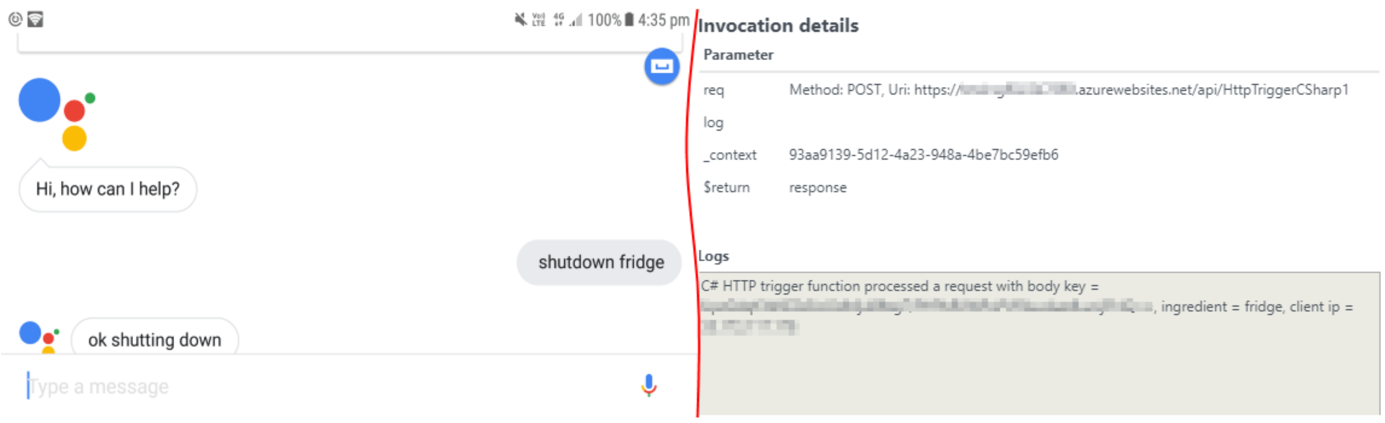
URL

https://[redacted].duckdns.org:8123/api/services/script/saugen?api_password=[redacted]

Surround any text with "<<<" and ">>>" to escape the content

Method

IN THE END HOW IT WOKS.



100% 4:35 pm

Hi, how can I help?

shutdown fridge

ok shutting down

Type a message

Invocation details

Parameter	
req	Method: POST, Uri: https://[redacted].azurewebsites.net/api/HttpTriggerCSharp1
log	
_context	93aa9139-5d12-4a23-948a-4be7bc59efb6
$return	response

Logs

```
C# HTTP trigger function processed a request with body key =
[redacted], ingredient = fridge, client ip =
[redacted]
```

Thank you!
grammarly